

D410

*data systems*



# NET Framework: Developing Applications Using ADO.Net ObjectSpaces

Luca Bolognese  
([lucabol@microsoft.com](mailto:lucabol@microsoft.com))  
Program Manager  
Microsoft Corporation

**Whidbey**

**PDC**<sup>03</sup>

Make the connection

**Microsoft**<sup>®</sup>

# WinFX Developer Preview

## Tools

Visual Studio .net  
Visual Basic .net  
Visual C++ .net  
Visual C# .net  
Visual J# .net

## Client Application Model

### Avalon

System.Windows

### Windows Forms

System.Windows.Forms

## Web & Service Application Model

### ASP.NET / Indigo

System.Web

## Data Systems Application Model

### Win FS

System.Storage

### Yukon

System.Data.SqlServer

## Mobile PC & Devices Application Model

### Compact Framework

System.Windows.Forms

### Mobile PC Optimized

System.Windows

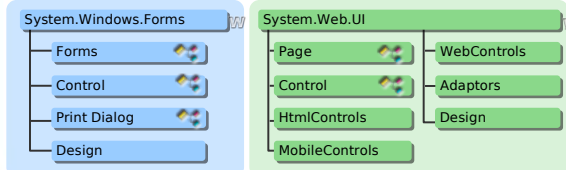
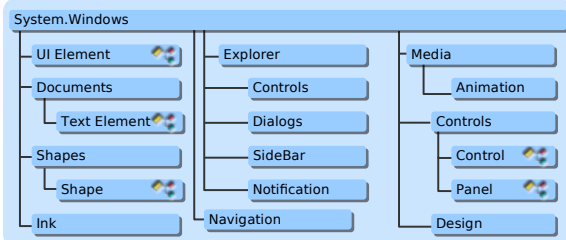
## Command Line

System.Console

## NT Service

System.ServiceProcess

## Presentation



System.Help

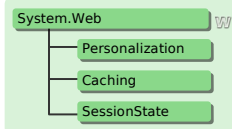
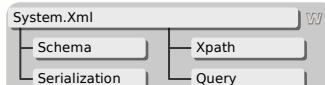
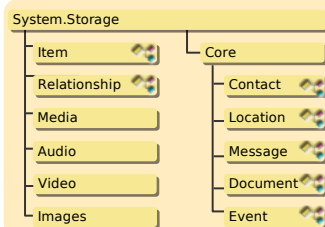
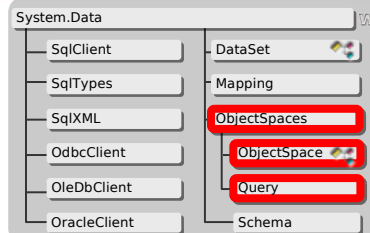
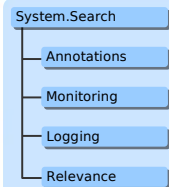
System.Drawing

System.NaturalLanguageServices

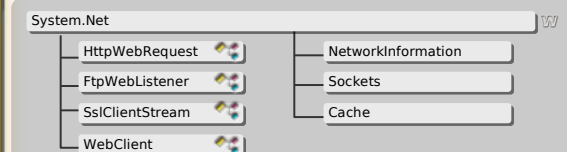
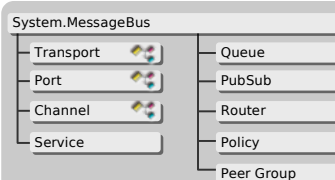
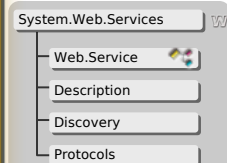
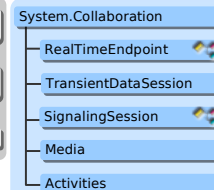
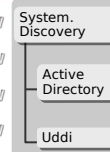
System.Speech

- Recognition
- Synthesis

## Data

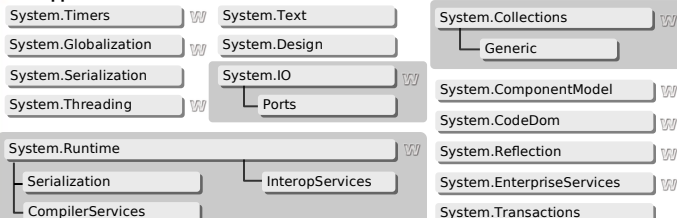


## Communication

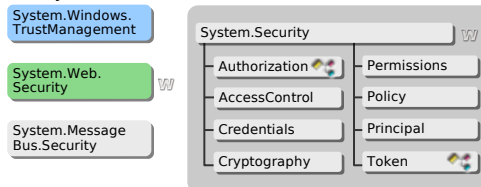


## Fundamentals

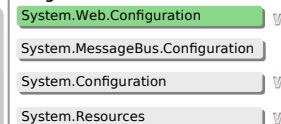
### Base & Application Services



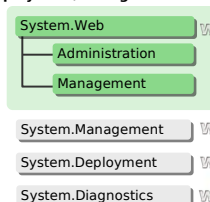
### Security



### Configuration



### Deployment/Management



# What Is It?

- Accessing and manipulating data in terms of domain objects: **Customer, Order, Address**
- Declarative mapping between these objects and relational tables
- Binary separation between business logic and data access logic
- Higher level layer on top of ADO.NET
- When does it make sense?
  - When you have a strong business logic layer

# ADO.NET Data Access

## Support

data is relational is objects is XML is Data

### Technology

### Strengths

### Use if...

#### **DataSet and DataReader in ADO.NET**

- Relational (tabular) model
- Highest performance
- Explicit control
- Fully exposes database functionality

- You are comfortable with the relational model
- You require maximum control/performance/functionality

#### **ObjectSpaces in ADO.NET**

- Business level objects
- Relational mapping via metadata
- Decoupled from database schema
- Smaller working set than DataSet

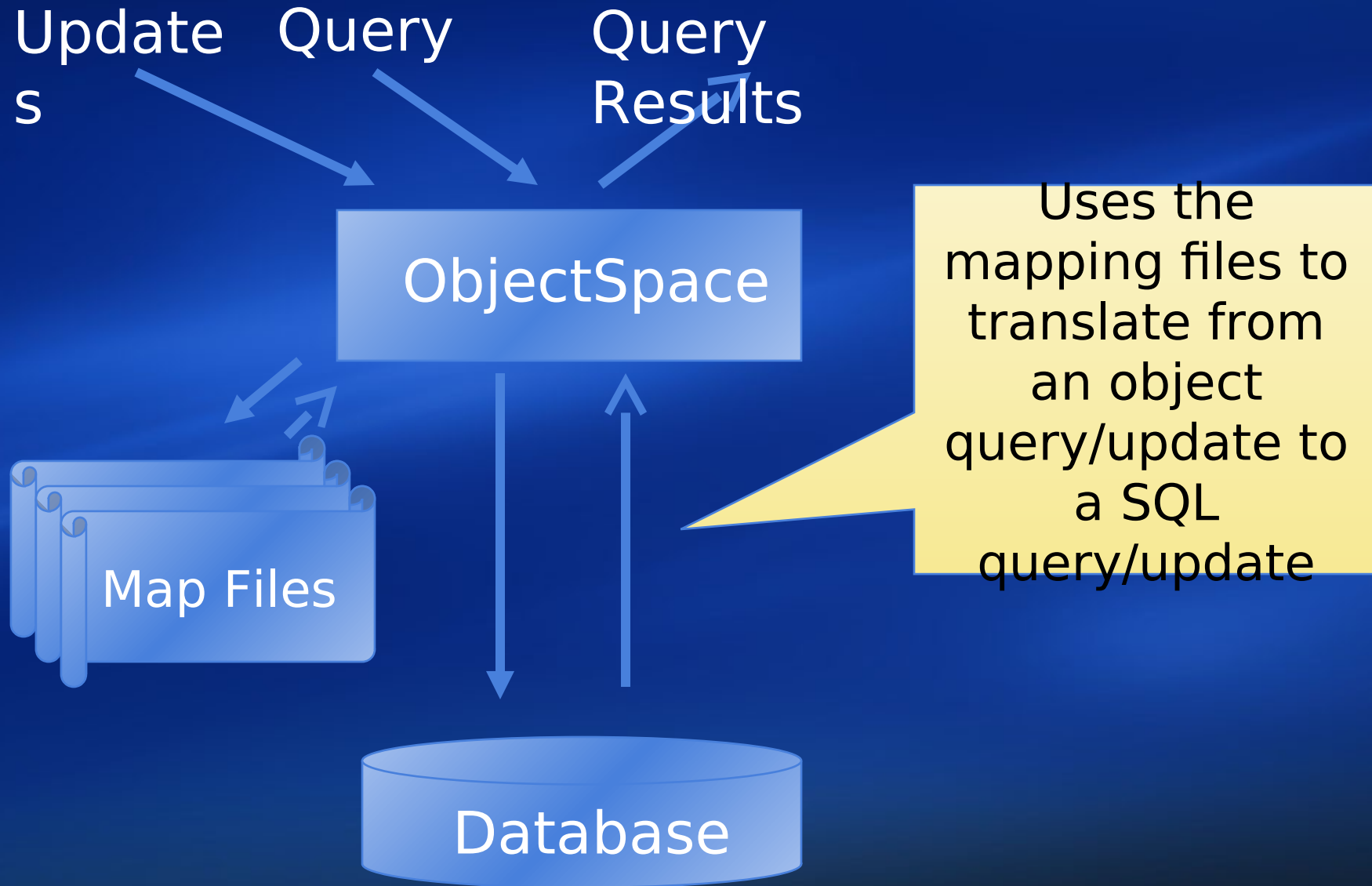
- You are using UI bound controls
- You need a strong business object layer
- You know the shape of the results you want to work with

#### **SQLXML in ADO.NET**

- Interoperability: Format for the Web - B2B, A2A
- Sparse (semi-structured) data
- XML Services e.g. XQuery, XSD
- Relational mapping via metadata
- Decoupled from database schema

- You need to query data from XML data sources e.g. XML Web Services
- You use vertical industry XML schemas for content publishing e.g. XBRL, RIXML, FinXML
- You need to load XML documents into database tables
- You are using UI bound controls for XML

# How Does It Work?





# How To Write Your Object Model And Initialization

## demo

Luca Bolognese  
Program Manager  
Webdata

**PDC**<sup>03</sup>

Make the connection

# How To Write Objects

- **Goal:** to be as transparent as possible
  - Like NFL referees...
- Normal CLR objects
  - Inheritance
  - 1-1, 1-many, many-many, methods etc...
- Current limitations
  - Need of an empty constructor (can be private)
  - ObjectList/ObjectHolder to achieve delay loading
- Initializing the database
  - Easier with the ObjectPersistence sample...

# OPath: How To Query For Objects

## demo

Luca Bolognese  
Program Manager  
Webdata

**PDC**<sup>03</sup>

Make the connection



# OPath

- Defined over the exposed object model
- Natural for the programmer
- Able to express complex queries
  - Relationships Navigation
  - Set Restriction Based navigation
- Examples querying for Customer
  - `Orders.Details.Quantity > 50`
  - `Orders[ShippedDate > RequiredDate]`
  - `Orders[Freight > 1000].Details.Quantity > 30`

# Span And Delay Load demo

Luca Bolognese  
Program Manager  
Webdata

**PDC**<sup>03</sup>

Make the connection

# Span And Delay Load

- How to materialize your object model
  - **Direct query**: multiple explicit queries for the right objects
  - **Span**: materializing connected objects when the primary ones are loaded
  - **Delay loading**: materializing connected objects just when they are accessed

# Updates, Transactions And Optimistic Concurrency

## demo

Luca Bolognese  
Program Manager  
Webdata

**PDC**<sup>03</sup>

Make the connection

# Updates, Transactions And Optimistic Concurrency

- Updates
  1. Modify your object model
  2. Call PersistentChanges
- Transactions
  - Wrap your code in BeginTransaction... CommitTransaction
- Optimistic concurrency: throw an exception if the data has changed in the data store. Alternatives:
  - Checking all the columns
  - Checking a subset of the columns
  - Using a timestamp field



# Visual Studio designer

**MapperDemo - Microsoft Development Environment**

File Edit View Project Build Debug Tools Window Help

Debug

**Manning1.msd**

AdventureWorks

- Classes
  - Company
    - Adr NW.Companies.Address
    - BusinessName NW.Companies.BusinessName
  - MyOrders
  - Item
  - Order
- Relationships
  - CompanyOrders
    - Company.MyOrders
    - Order.OrderCompany

**NW**

- Tables
  - Companies(Companies)
    - Address
    - BusinessName**
    - CompanyID
  - Items(Items)
  - OrderDetails(OrderData)
    - ItemID
    - OrderID
  - Orders(Orders)
    - CompanyID
    - OrderID

**Solution Explorer - Solution 'Map...**

Solution 'MapperDemo' (1 project)

- MapperDemo
  - References
  - AssemblyInfo.cs
  - Class1.cs
  - Mapping1.msd
  - osd.xml
  - osmcode.cs
  - rsd.xml

**Solution Explorer - Class View**

Properties

**BusinessName** DTCColumn

**Misc**

Name BusinessName

**Name**

**Task List**

!	Description	File	Line
	Click here to add a new task		

Item(s) Saved

Start MapperDe...

4:22 PM

# Sample Mapping Tool demo

Luca Bolognese  
Program Manager  
Webdata

**PDC**<sup>03</sup>

Make the connection

# ADO.NET Interoperability

## demo

Luca Bolognese  
Program Manager  
Webdata

**PDC**<sup>03</sup>

Make the connection

# What I Haven't Shown You...

- Complex mappings
  - Inheritance
  - Many to many, one object to multiple table etc...
- ObjectEngine: the low level interface
- Implicit management of primary key
- And much more...

# Future: Using Generics

```
ObjectQuery<Customer> query =  
    new ObjectQuery<Customer>("State = 'WA'", "Orders");  
  
ObjectReader<Customer> or =  
    objectspace.GetObjectReader(query);  
  
or.Read();  
Customer customer = or.Current;
```

```
Dim query As New _  
    ObjectQuery(Of Customer)("State = 'WA'", "Orders")  
  
Dim or As ObjectReader(Of Customer) = _  
    objectspace.GetObjectReader(query)  
  
or.Read()  
Dim cust As Customer = or.Current
```



# A Look Into The Future...

- Using generics in the framework
  - Pointer to generics talk
- Enhancing OPath
  - New operators and functions in the language
- A report query language
  - How to get just some properties of your object
- DML Delete
  - How to delete some objects without retrieving them first

# Conclusions

- **Everything in my demo works with your bits!!!**
- It is especially useful when you have a lot of business logic in your application
- It is not intrusive for the object model
- It is not prescriptive for the programming model
- It is included in the next version of Visual Studio

# Community Resources

## Get Your Questions Answered!

- Newsgroups:
  - microsoft.public.objectspaces
- Webdata booth
  - connect with Microsoft product teams, and PDC 2003 Speakers
- Ask The Experts:
  - Tuesday 7 pm – 9 pm in Hall G,H
- Email:
  - [osfdbck@microsoft.com](mailto:osfdbck@microsoft.com)

